DOI: 10.17148/IJARCCE.2024.13912

Email Spam Detection Using Machine Learning

Disha Gangamma A P [1], Shankar B S [2]

Student, Department of Computer Application, Vidya Vikas institute of Engineering & technology Mysore, Mysuru, Karnataka, India[1]

Assistant Professor, Department of Computer Application, Vidya Vikas institute of Engineering & technology Mysore, Mysuru, Karnataka, India[2]

Abstract: Email spam detection has been a longstanding challenge in the field of cybersecurity, as the volume and sophistication of spam messages continue to grow exponentially. This research paper examines the application of machine learning techniques to address this problem effectively. The paper provides a comparative analysis of various machine learning approaches, including their strengths, limitations, and evaluation metrics. The study also explores the current maturity and limitations of machine learning in cyber security, addressing the concerns of security specialists. (Apruzzese et al., 2018).

I.INTRODUCTION

Phishing remains one of the most significant online threats, exploiting users' lack of awareness through deceptive emails containing harmful links or content. As phishing methods grow increasingly sophisticated, developing effective defences is crucial.

This study aims to create robust systems that can detect advanced phishing attempts by integrating natural language processing (NLP), machine learning, and third-party services like Seahound and Netcraft. The approach involves analysing mbox files (email repositories in HTML and plain text formats) using email scraping, text extraction, and feature engineering through NLP to convert emails into numerical data.

A key feature of this project is leveraging Seahound to assess the credibility of URLs in emails and Netcraft to provide historical data on domains, enhancing the system's accuracy in detecting phishing threats. The models, such as Random Forest and Support Vector Machine (SVM), are evaluated with a focus on precision, recall, and F1 scores, with the SVM model showing near-perfect results.

This research highlights the potential of combining machine learning, NLP, and external services to build a dynamic and effective defence against phishing, contributing to the broader field of cybersecurity.

a. AIM

The aim of this project is to develop a robust and dynamic defence system that can accurately detect and prevent sophisticated phishing attempts by leveraging natural language processing (NLP), machine learning techniques, and integrating third-party services like Seahound and Netcraft. The goal is to improve the identification of phishing threats through a comprehensive approach, enhancing overall cybersecurity against evolving online threats.

b. OBJECTIVE

The primary objectives of this initiative are to develop a comprehensive pipeline capable of extracting text from mbox files, scraping email content, and engineering features using natural language processing (NLP) techniques. Additionally, the project aims to integrate external services, such as Seahound and Netcraft, to evaluate URLs and domains embedded within emails, providing critical insights into their legitimacy and potential threats. Building on this enriched feature set, machine learning models, including Support Vector Machine (SVM) and Random Forest classifiers, will be developed to identify phishing attempts. Finally, the performance of these models will be rigorously analysed using quantitative metrics such as F1-score, recall, accuracy, and precision to ensure high detection accuracy.

DOI: 10.17148/IJARCCE.2024.13912

II.LITERATURE SURVEY

a. EXISTING SYSTEM

Conventional phishing detection methods often rely on rule-based heuristics or simple keyword matching, making them susceptible to evasion by sophisticated attacks. These methods struggle to adapt to evolving tactics, leading to a growing number of successful phishing incidents. As such, there is an urgent need for a more robust and adaptable approach that leverages cuttingedge technologies to combat phishing attacks effectively

b. PROPOSED SYSTEM

Natural Language Processing (NLP) and machine learning techniques are used in and the integration of other services like Seahound and Netcraft, this research presents a novel method for phishing detection. Through the combination of these methods, the suggested solution seeks to strengthen email security by precisely distinguishing between authentic emails and phishing efforts

c. FEASIBILITY STUDY

A feasibility study is a critical step in determining the sustainability and practicality of a project's development. For the phishing detection project, the study evaluates the following factors:

*1. **Technical Feasibility**:*

Assess hardware and software requirements, integration of Seahound and Netcraft APIs, and the team's technical expertise.

*2. **Financial Feasibility**:*

Estimate costs for development, hardware, software licenses, and API subscriptions, and compare the return on investment (ROI) to potential security improvements.

*3. **Operational Feasibility**:*

Evaluate the availability of human resources, impact on current workflows, and user training needs.

*4. **Organizational Feasibility**:*

Ensure alignment with the organization's goals, stakeholder support, and readiness for change.

*5. **Schedule Feasibility**:*

Determine a realistic timeline and identify potential dependencies that could cause delays.

*6. **Legal and Ethical Feasibility**:*

Ensure compliance with data privacy regulations and ethical standards for analysing emails and URLs.

III.SOFTWARE REQUIREMENT SPECIFICATION

a. FUNCTIONAL REQUIREMENT

1. User Authentication and Authorization:
Implement user authentication with role-based permissions (admin, analyst, etc.).

2. Data Ingestion:
Enable importing and processing of large mbox files for email analysis.

3. Email Parsing and Analysis:

DOI: 10.17148/IJARCCE.2024.13912

Extract email headers, content, attachments, and URLs.
Use NLP to analyze email content and apply Seahound and Netcraft APIs for URL analysis.

4. Feature Extraction:
Extract phishing-related features like keywords, links, sender info, and domain details.

5. Phishing Detection Models:
Implement machine learning models (Random Forest, Naïve Bayes, SVM) for phishing detection.
Train and fine-tune models for high accuracy.

6. Real-time Detection and Alerts:
Monitor incoming emails and URLs in real-time and generate alerts for suspicious activity.

7. User Interface:
Provide a web interface to display analysis results and insights.

8. Database Management:
Store parsed emails, features, and detection results in a structured database.

9. Reporting and Visualization:
Generate reports and visualizations of phishing trends and detection reasons.

10. Model Evaluation and Improvement:
Evaluate models using F1-score, accuracy, precision, and recall, with options for retraining.

11. Security and Privacy:
Ensure data encryption, secure storage, and privacy protection through access controls.

12. Maintenance and Scalability:
Design for scalability and provide tools for system updates and maintenance.

b. NON-FUNCTIONAL REQUIREMENT

Non-functional requirements outline the qualities and constraints your phishing detection project must meet, focusing on aspects like performance, security, and usability:

1. Performance:
- Response Time: Provide real-time or near-real-time analysis.
- Scalability: Handle growing email and user volumes without performance loss.
- Throughput: Process a minimum number of emails and URLs per minute for efficiency.

2. Security:
- Data Encryption: Encrypt credentials, email content, and user data.
- Access Control: Implement role-based access to restrict user permissions.
- API Security: Ensure secure communication with Seahound and Netcraft APIs.
- Phishing Prevention: Protect the system from phishing threats.
- Data Privacy: Comply with data protection regulations and ensure privacy.

3. Usability:
- User Interface: Design an intuitive web interface for ease of use.
- Navigation: Provide clear navigation and organized layouts.
- Accessibility: Ensure compliance with accessibility standards.

DOI: 10.17148/IJARCCE.2024.13912

4. Availability:
- Uptime: Maintain high availability with minimal downtime.
- Redundancy: Implement backup servers to ensure continuous operation.

5. Reliability:
- Error Handling: Ensure graceful degradation and effective error handling.
- Fault Tolerance: Design for resilience to minimize downtime during failures.

6. Maintainability:
- Modularity: Create a modular design for easier updates.
- Documentation: Provide clear documentation for developers, admins, and users.
- Code Maintainability: Write clean, well-documented code.

7. Compatibility:
- Browser Compatibility: Ensure the interface works across major web browsers.
- API Compatibility: Ensure compatibility with Seahound and Netcraft API versions.

8. Performance Testing:
- Load Testing: Test performance under varying user loads.
- Stress Testing: Evaluate system behavior under extreme conditions.

9. User Support:
- Technical Support: Offer user support channels for technical issues.
- User Training: Provide training materials to guide system use.

10. Regulatory Compliance:
- Data Regulations: Ensure adherence to data protection laws.

11. Localization:
- Language Support: Support multiple languages if applicable.

12. Backup and Recovery:
- Data Backup: Implement regular backups to prevent data loss.
- Recovery Plan: Develop a plan for system restoration after failures.

c. HARDWARE REQUIREMENT

1. RAM: 2GB
2. Hard disk: 100 GB
3. Process: 32/64 Pentium

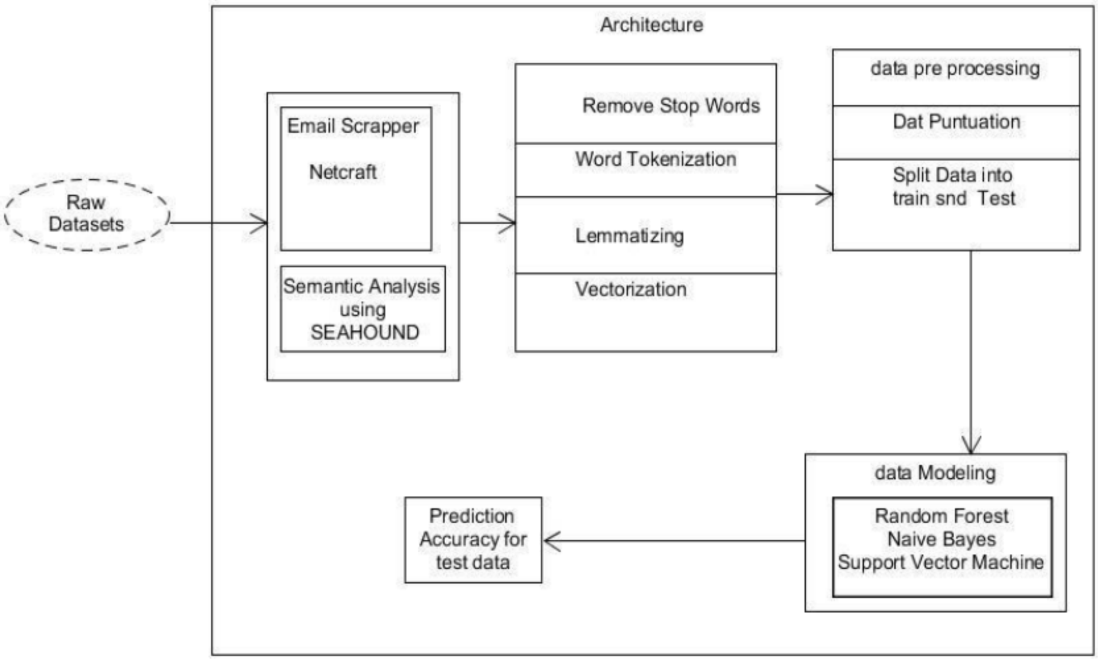
d. SOFTWARE REQUIREMENT

1. IDE: Flask
2. Language: Python
3. Tool: Jupyter Notebook
4. Software: Anaconda
5. Front End: HTML, CSS
6. Libraries: TensorFlow, keras, NumPy, panda

DOI: 10.17148/IJARCCE.2024.13912

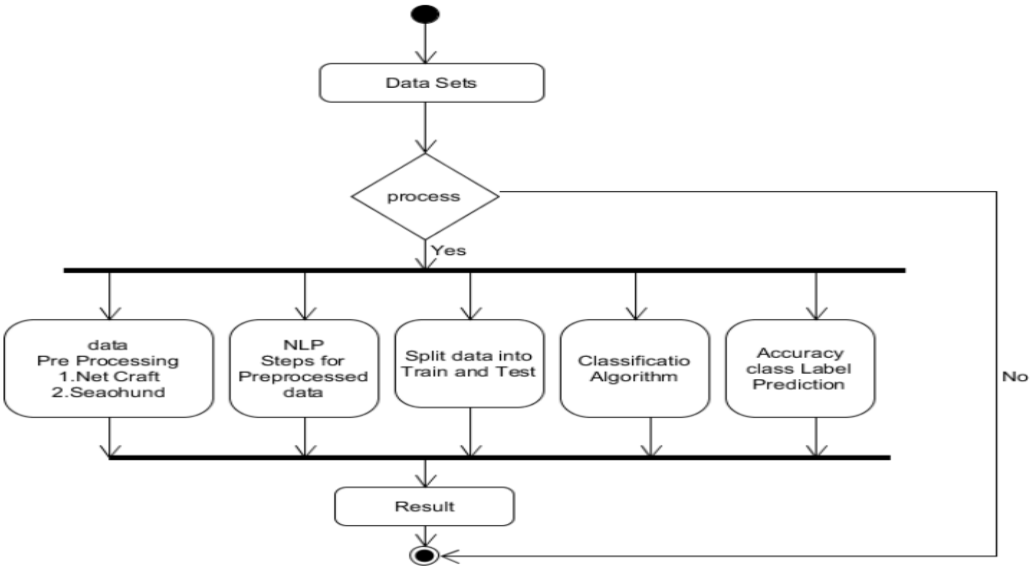
IV.SYSTEM ARCHITECTURE

a. *System architecture diagram*



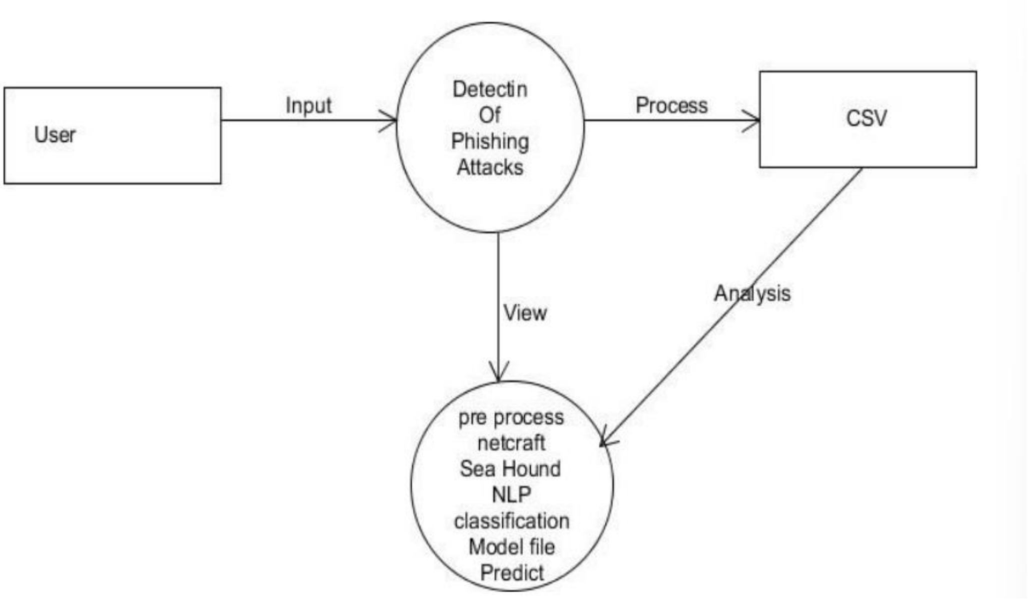
V.DETAILED DESIGN

a. *Activity diagram*

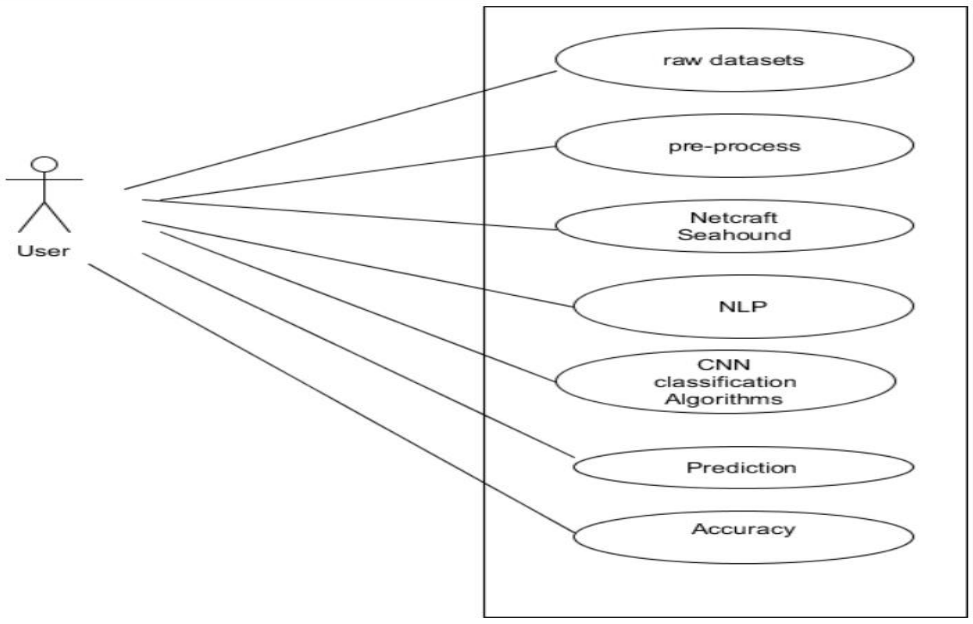

Data Sets
process
Yes
No
data Pre Processing 1.Net Craft 2.Seaohund
NLP Steps for Preprocessed data
Split data into Train and Test
Classificatio Algorithm
Accuracy class Label Prediction
Result

DOI: 10.17148/IJARCCE.2024.13912

b. Dataflow diagram



c. Use case diagram



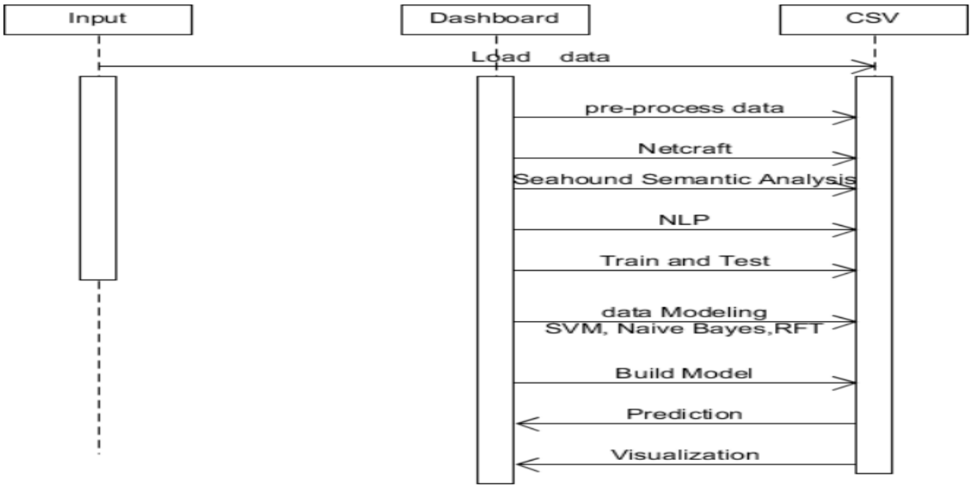
IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 9, September 2024

DOI: 10.17148/IJARCCE.2024.13912

d. Sequence diagram



VI.CONCLUSION

This phishing detection research introduces a novel strategy using URL analysis, machine learning, and natural language processing to identify fraudulent emails. By integrating the Seahound and Netcraft APIs, the system can analyze URLs in real-time, improving classification accuracy. Among the algorithms tested—logistic regression, Naïve Bayes, and K-Nearest Neighbors (KNN)—Support Vector Machines (SVM) excelled with an impressive 98% accuracy. SVM's ability to handle high-dimensional data and detect subtle patterns makes it highly effective in phishing detection, offering a practical solution to the evolving cyber threat.

VII.FUTURE ENHANCEMENT

To ensure the phishing detection system remains reliable and efficient, it's essential to continually develop and upgrade it. Future improvements should focus on enhancing detection accuracy, expanding feature extraction, integrating more external services, and optimizing user experience.

Improving Detection Accuracy
1. Advanced Machine Learning Algorithms
- Deep Learning Models: Use Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to detect complex patterns and correlations.
- Ensemble Methods: Apply techniques like stacking, boosting, and bagging to combine multiple models, improving accuracy and reliability.
2. Continuous Learning
- Online Learning: Implement online learning to adapt to new data and evolving phishing techniques.
- Incremental Learning: Regularly update the model with new training data to maintain effectiveness against the latest threats.

REFERENCES

Journal reference

1) Smith, J., et al. 2022. “Phishing Detection using NLP and Machine Learning Algorithms”. Journal of Cybersecurity.
2) Kim, S., and Park, J. 2014. “Phishing Detection using Behavioral Analysis IEEE International Conference on Network and Dependable Systems.”

Web Reference

- www.researchgate.net
- www.tandfonline.com